

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1 -5. (Cancelled)

6. (Currently Amended) The method of claim 31 further comprising providing a cursor on any type of query executed.

7. (Currently Amended) The method of claim 31 wherein a programming model for said an out-of-process application is symmetrical with an in-process programming model for said the DBMS.

8. (Currently Amended) The method of claim 31 further comprising the marshaling of data between an unmanaged layer and a managed layer.

9. (Currently Amended) The method of claim 31 wherein an application operation from a group of operations comprising functions, procedures, and triggers is executed directly in the DBMS.

10. (Currently Amended) The method of claim 9 wherein a result is returned by said the DBMS to said application a client based on the execution of said the application operation by said the DBMS.

11 -15. (Cancelled)

16. (Currently Amended) The system of claim 14 37 further comprising a subsystem for providing a cursor on any type of query executed.

17. (Currently Amended) The system of claim 14 37 wherein a programming model for said an out-of-process application is symmetrical with an in-process programming model for said the DBMS.

18. (Currently Amended) The system of claim ~~14~~ 37 further comprising a subsystem for the marshaling of data between an unmanaged layer and a managed layer.
19. (Currently Amended) The system of claim ~~14~~ 37 further comprising a subsystem for an application operation from a group of operations comprising functions, procedures, and triggers to be executed directly in the DBMS.
20. (Currently Amended) The system of claim 19 further comprising a subsystem by which a result is returned by said the DBMS to said application a client based on the execution of said the application operation by said the DBMS.
- 21-25. (Canceled)
26. (Currently Amended) The computer-readable instructions of claim ~~24~~ 43 further comprising instructions for providing a cursor on any type of query executed.
27. (Currently Amended) The computer-readable instructions of claim ~~24~~ 43 further comprising instructions for a programming model for said an out-of-process application that is symmetrical with an in-process programming model for said the DBMS.
28. (Currently Amended) The computer-readable instructions of claim ~~24~~ 43 further comprising instructions for the marshaling of data between an unmanaged layer and a managed layer.
29. (Currently Amended) The computer-readable instructions of claim ~~24~~ 43 further comprising instructions for an application operation from a group of operations comprising functions, procedures, and triggers to be executed directly in the DBMS.

30. (Currently Amended) The computer-readable instructions of claim 29 further comprising instructions whereby a result is returned by ~~said the~~ DBMS to ~~said application a~~ client based on the execution of ~~said the~~ application operation by ~~said the~~ DBMS.

31. (New) A computer-implemented method for executing .NET managed code in a database management system (DBMS) having a database server, the method comprising:
invoking .NET managed code and an invocation context in the database server;
exposing the invocation context to the database server through the utilization of an in-process provider; and
executing the .NET managed code in the database server based on the invocation context.

32. (New) The method of claim 31, wherein exposing the invocation context comprises exposing at least one of:
a client's connection context,
a command with a state execution context;
a transaction context associated with a command;
a path through which requests and results may be sent or received between a client and database server;
a trigger context, wherein the trigger results from an operation of the client; or
a forward-only cursor on top of statement execution results.

33. (New) The method of claim 31, further comprising a client, wherein the client is a .NET application and the in-process provider is an ADO.net in-process provider.

34. (New) The method of claim 31, further comprising separating the .NET managed code into an immutable part and a mutable part and, and executing the .NET managed code based on the results of the operation of separating.

35. (New) The method of claim 31, further invoking .NET managed code in the database server as a result of a client trigger.

36. (New) The method of claim 31, wherein the in-process provider supports more than one pending executing command for a client connection.

37. (New) A system for executing application code in a database management system (DBMS) comprising a processor and a memory, the system comprising:

 a subsystem for invoking .NET managed code and an invocation context in the database server;

 a subsystem for exposing the invocation context to the database server through the utilization of an in-process provider; and

 a subsystem for executing the .NET managed code in the database server based on the invocation context.

38. (New) The system of claim 37, wherein exposing the invocation context comprises exposing at least one of:

 a client's connection context,

 a command with a state execution context;

 a transaction context associated with a command;

 a path through which requests and results may be sent or received between a client and database server;

 a trigger context, wherein the trigger results from an operation of the client; or

 a forward-only cursor on top of statement execution results.

39. (New) The system of claim 37, further comprising a client subsystem, wherein the client subsystem comprises a .NET application, and wherein the in-process provider is an ADO.net in-process provider.

40. (New) The system of claim 37 further comprising a subsystem for separating the .NET managed code into an immutable part and a mutable part and, and a subsystem for executing the .NET managed code based on the results of the operation of separating.

41. (New) The system of claim 37, wherein invoking .NET managed code in the database server is a result of a client trigger.

42. (New) The system of claim 37, wherein the in-process provider supports for more than one pending executing command for a client connection.

43. (New) A computer-readable storage medium comprising computer-readable instructions for executing application code in a database management system (DBMS), the computer-readable instructions comprising instructions for:

invoking .NET managed code and an invocation context in the database server;
exposing the invocation context to the database server through the utilization of an in-process provider; and
executing the .NET managed code in the database server based on the invocation context.

44. (New) The computer-readable instructions of claim 43, wherein exposing the invocation context comprises exposing at least one of:

a client's connection context,
a command with a state execution context;
a transaction context associated with a command;
a path through which requests and results may be sent or received between the client and database server;
a trigger context, wherein the trigger results from an operation of the client; or
a forward-only cursor on top of statement execution results.

45. (New) The computer-readable instructions of claim 43, further comprising a client, wherein the client comprises a .NET application, and wherein the in-process provider is an ADO.net in-process provider.

46. (New) The computer-readable instructions of claim 43, further comprising instructions for separating the .NET managed code into an immutable part and a mutable part

and, and instructions for executing the .NET managed code based on the results of the operation of separating.

47. (New) The computer-readable instructions of claim 43, wherein invoking .NET managed code in the database server is a result of a client trigger.

48. (New) The computer-readable instructions of claim 43, wherein the in-process provider supports more than one pending executing command for a client connection.